

### **Remarks and Arguments**

The specification has been objected to because the abstract exceeds the limit of 150 words. A new abstract has been provided which does not exceed the limit.

Claims 19-32, 56-69, 77 and 81 have been submitted for examination. Claims 21-24, 27, 30-32, 58-61, 64, 67-69, 77 and 81 have been amended. Claims 19-20 and 56-57 have been canceled.

Claims 19-32, 56-69, 77 and 81 were rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 6,442,570 (Wu.) The examiner asserts that Wu discloses all of the claimed limitations with the exception that Wu does not explicitly disclose "insuring that the first and second data are continually equated." however the examiner claims that Wu teaches that two objects are synchronized and that it would have been obvious to apply the teachings of Wu to make the objects equivalent.

The present invention is addressed to the problem of storing XML objects that are concurrently being created and modified by two separate processes running on the same device. Since each process could be concurrently attempting to examine and change the data in these objects, the integrity of the data in the objects can be compromised if both processes attempt to modify the same object at the same time. The present invention uses a single data region to store all of the data and this single region is mapped into the address spaces used by each of the separate processes. A locking mechanism is used to prevent data corruption - each process would need to lock the region for write or read. This lock would prevent the other process from modifying the region while the write or read operation was going on.

In contrast, Wu deals with the synchronization of two separate devices (for example, a desktop computer and portable computer, See the Wu abstract). These devices store data in two separate data stores (Wu, column 3, line 67-column 4, line 3.) This is a fundamentally different problem than the problem solved by the invention. Consequently, Wu does not disclose any mechanism to lock the data. In addition, it would not be possible to modify the Wu synchronization manager to add such a locking mechanism because Wu's synchronization manager is not integrated with the applications that actually change the data. The examiner claims that column 6, lines 50-59 of the Wu patent teach locking, but the quoted section just describes how the

synchronization process is separate from the applications. For example, Wu states that the portable synchronization manager and the desktop synchronization manager do not interpret or understand the data entries contained within the synchronized objects (lines 52-55.) See also Wu column 6, lines 50-52. Rather than teaching locking, this is the reason that Wu cannot lock the data. Thus, an application in the desktop computer and an application in the portable computer can change the same data. If this happens, the user must resolve the conflict (see Wu, column 4, lines 9-11.)

The examiner also cites column 9, lines 22-44, as teaching locking. However, this section describes how the objects that need to be synchronized are identified and is not concerned with locking. This section describes using a FindObjects function that is called by the portable synchronization manager to locate objects in the portable computer that have been changed since the last synchronization. Locking of the found objects is not described as this locking would require knowledge of the object data, which the synchronization manager does not have (Wu column 6, lines 52-55.)

The claims have been amended to particularly point out the inventive locking mechanism. For example, amended claim 22 recites a first storage manager that constructs a document object including data stored in a region (lines 7-10), a second storage manager that constructs another document including data stored in the same region (lines 11-15) and a synchronization manager that locks the region data when the first process is changing the region data in the first address space (lines 16-17.) From the discussion above, it is clear that the Wu synchronization apparatus does not store all of the data in a single region and does not have any region data locking capability. Consequently, Wu cannot teach or suggest the claimed combination and amended claim 22 patentably distinguishes thereover.

Claims 21 and 23-32 are dependent, either directly or indirectly, on amended claim 22. Therefore, they incorporate the limitations of amended claim 22 and patentably distinguish over the Wu reference in the same manner as amended claim 22. In addition, these claims recite further limitations not taught or suggested by Wu. For example, claim 23 recites that the second process comprises methods for requesting that the synchronization manager lock the region data when the second process is changing the region data in the second address space. As discussed above Wu, does

not teach or suggest locking. Similarly, claim 25 recites that the first process comprises methods for requesting that the synchronization manager lock the region data during the processing of all read and write operations in a transaction.

Further claim 30 recites that the synchronization mechanism comprises a distributed memory system. The examiner states that a distributed memory system is disclosed at Wu column 5, lines 38-45. This section of the Wu reference describes using a communications link to communicate between the desktop computer and the portable computer (see Wu column 5, lines 46-61.) The "distributed memory" recited in claim 30 is described in the instant specification at page 22, line 10 to page 26, line 22. As described therein, a distributed memory system allows data regions in separate devices to be synchronized. A communication link would not be considered a distributed memory because it cannot create, map and unmap data regions.

Claim 32 recites that the storage managers each construct a "cross process synchronization object." This object, which is basically a lock, is described in the present specification at page 39 and is used to coordinate between the two processes. The examiner equates this to an "application synchronization module" described by Wu at column 4, lines 46-59. As described by Wu, the application synchronization module is program code that has knowledge of the objects used by the applications. This knowledge is used in the synchronization process.

Claim 59 has been amended in a manner similar to amended claim 22 and distinguishes over the Wu reference in the same manner as amended claim 22.

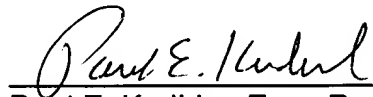
Claims 58 and 60-69 are dependent, either directly or indirectly, on amended claim 59. Therefore, they incorporate the limitations of amended claim 59 and patentably distinguish over the Wu reference in the same manner as amended claim 59. In addition, these claims have similar scope to claims 21 and 23-32 discussed above and recite further limitations not taught or suggested by Wu.

Claims 77 and 81 have been amended in a manner similar to amended claim 22 and distinguish over the Wu reference in the same manner as amended claim 22.

In light of the forgoing amendments and remarks, this application is now believed in condition for allowance and a notice of allowance is earnestly solicited. If the examiner has any further questions regarding this amendment, he is invited to call

applicants' attorney at the number listed below. The examiner is hereby authorized to charge any fees or direct any payment under 37 C.F.R. §§1.17, 1.16 to Deposit Account number 02-3038.

Respectfully submitted



Date: 7/2/04

Paul E. Kudirka, Esq. Reg. No. 26,931

KUDIRKA & JOBSE, LLP

Customer Number 021127

Tel: (617) 367-4600 Fax: (617) 367-4656